Amendments to the Claims:

1. (Currently Amended) An apparatus comprising a computing device programmed with an extensible framework that configures the computing device to accept ~~accepts~~ one or more mark-up language parsers ~~and/~~or generators, each implemented as plug-ins to the framework, ~~with~~ wherein different plug-ins configure the device to handle ~~enabling~~ different kinds of mark up languages ~~to be handled by the device~~;

wherein the device is further programmed with a generic data supplier application programming interface (API) and wherein said parsers or generators configure the device to access data from said data supplier API; and

wherein said generic data supplier API configures the device to access data from at least one data source and decouple said parsers or generators from said at least one data source.

2. (Currently Amended) The apparatus ~~computing device~~ of Claim 1 in which the extensible framework (i) insulates a client running on the device from having to communicate directly with at least one of the one or more parsers or generators~~a parser or generator~~ and (ii) is generic in that it presents a common API to the client irrespective of the specific kind of parser or generator deployed.

3. (Currently Amended) The ~~computing device~~apparatus of Claim 1 in which ~~the~~ a client interacts with several kinds of parser or generator plug-ins to the extensible framework, each handling different mark-up language formats.

4. (Currently Amended) The ~~computing device~~apparatus of Claim 1, wherein the computing device is programmed with a file conversion capability requiring (i) a source file to be parsed by the parser, which is adapted to handle one format and (ii) an output, converted file to be generated by one or more of the generators, adapted to handle a different file format.

5. (Currently Amended) The ~~computing device~~apparatus of Claim 1 in which several different clients are able to share the same parsers or generators.

6. (Currently Amended) The ~~computing device~~apparatus of Claim 1 in which the extensible framework enables at least one of the one or more ~~a~~ parsers or generators to access data from any source that conforms to ~~a~~ the generic data supplier API.

7. (Currently Amended) The ~~computing device~~apparatus of Claim 6 in which the source is a buffer in memory.

8. (Currently Amended) The apparatus~~computing device~~ of Claim 6 in which the source is a file.

9. (Currently Amended) The apparatus~~computing device~~ of Claim 6 in which the source is a socket outputting streaming data.

10. (Currently Amended) The apparatus~~computing device~~ of Claim 6 in which the computing device is configured to use ~~uses a~~ data source different from any data sources which the device was capable of using when first operated by an end-user.

11. (Currently Amended) The apparatus~~computing device~~ of Claim 1, in which the extensible framework configures the computing device to enable ~~enables~~ the mark-up language parser or generator to access components to validate, pre-filter or alter data, in which the components are plug-in components to the framework and configure the computing device to operate using a chain of responsibility design pattern.

12. (Currently Amended) The apparatus~~computing device~~ of Claim 11 in which the plug-in components configure the computing device to ~~all~~ present a common, generic API to the parser or generator, enabling the same plug-in components to be used with different types of parsers and generators.

13. (Currently Amended) The apparatus~~computing device~~ of Claim 11 in which the plug-in components ~~all~~ configure the computing device to present a common, generic API to a client component using the parser or generator, enabling the same plug-in components to be used by different clients.

14. (Currently Amended) The apparatus~~computing device~~ of Claim 11 in which the parser configures the computing device to notify~~notifies~~ a validator plug-in of elements ~~it is parsing~~being parsed and these in turn go to an auto correction plug-in to be fixed if required and finally a client receives these events.

15. (Currently Amended) The apparatus~~computing device~~ of Claim 11 in which a parsed element stack is made available to all validation/pre-filter/altering plug-ins.

16. (Currently Amended) The apparatus~~computing device~~ of Claim 11 in which the computing device incorporates a character conversion module that enables documents written in different character sets to be parsed and converted to a common, generic character set.

17. (Currently Amended) The apparatus~~computing device~~ of Claim 1 in which extensions to ~~its~~ capabilities of the computing device are~~can be~~ made without affecting compatibility with existing clients or existing parsers and generators by the use of an updated/extended namespace plug-in that sets-up ~~all the~~ elements, attributes and attribute values for a namespace.

18. (Currently Amended) A method ~~of parsing a mark-up language document,~~ comprising: ~~the step of~~

accessing, via a computing device, an extensible framework that accepts parser or generator plug-ins, with different parser plug-ins or generator plug-ins enabling different kinds of mark up languages to be handled;

enabling said parser plug-ins or generator plug-ins to access data from a generic data supplier application programming interface (API); and

enabling said generic data supplier API to access data from at least one data source;

wherein the extensible framework and the generic data supplier API are loaded on a computing device, and said generic data supplier API decouples said parser plug-ins or generator plug-ins from said at least one data source.

19. (Cancelled)

20. (Currently Amended) The method of Claim 18, further comprising extending ~~in which extensions to device~~ capabilities of a device configured to access the extensible framework ~~can be made~~ without affecting compatibility with existing clients or existing parsers and generators by the use of an updated/extended namespace plug-in that sets-up ~~all the~~ elements, attributes and attribute values for a namespace.

21. (Previously Presented) The method of Claim 18, in which the extensible framework (i) insulates a client running on ~~the~~ a device configured to access the extensible framework from having to communicate directly with a parser or generator and (ii) is generic in that it presents a common API to the client irrespective of the specific kind of parser or generator deployed.

22. (Currently Amended) The method of Claim 18, further comprising interacting ~~in which the client interacts~~ with several kinds of parser plug-ins or generator plug-ins to the extensible framework via a client, each handling different mark-up language formats.

23. (Currently Amended) The method of Claim 18, further comprising implementing a ~~in which the device is programmed with a~~ file conversion capability to parse ~~requiring (i)~~ a source file ~~to be parsed~~ by the parser, which is adapted to handle one format and ~~(ii)~~ cause an output, converted file to be generated by the generator, adapted to handle a different file format.

24. (Previously Presented) The method of Claim 18, in which several different clients are able to share the same parsers or generators.

25. (Currently Amended) The method of Claim 18, further comprising ~~in which the extensible framework enables~~enabling, via the extensible framework, a parser or generator to access data from any source that conforms to ~~a~~the generic data supplier API.

26. (Original) The method of preceding Claim 25, in which the source is a buffer in memory.

27. (Original) The method of preceding Claim 25 in which the source is a file.

28. (Original) The method of preceding Claim 25 in which the source is a socket outputting streaming data.

29. (Previously Presented) The method of Claim 25, in which a data source is used that is different from any data sources which the device was capable of using when first operated by an end-user.

30. (Currently Amended) The method of Claim 18, further comprising enabling, via ~~in which~~ the extensible framework, ~~enables~~ the mark-up language parser or generator to access components to validate, pre-filter or alter data, in which the components are plug-in components to the framework and operate using a chain of responsibility design pattern.

31. (Currently Amended) The method of preceding Claim 30, in which the plug-in components ~~all~~ present a common, generic API to the parser or generator, enabling the same plug-in components to be used with different types of parsers ~~and~~ or generators.

32. (Currently Amended) The method of Claim 30, in which the plug-in components ~~all~~ present a common, generic API to a client component using the parser or generator, enabling the same plug-in components to be used by different clients.

33. (Currently Amended) The method of Claim 30, ~~in which~~further comprising causing, via the parser, ~~notifies~~ a validator plug-in to be notified of elements ~~it~~the parser is parsing, the elements ~~and these~~ in turn going to an auto correction plug-in to be fixed if required and finally to a client that receives these elements~~events~~.

34. (Currently Amended) The method of preceding Claim 30, further comprising making ~~in which~~ a parsed element stack ~~is made~~ available to ~~all~~ validation/pre-filter/altering plug-ins.

35. (Currently Amended) The method of Claim 30, enabling, via ~~in which~~ a character conversion module, ~~enables~~ documents written in different character sets to be parsed and converted to a common, generic character set.